Problem A. At the midpoint

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Your team is participating in the Argentinian Programming Tournament, a contest that lasts exactly 5 hours in total.

In your head, your coach's voice echoes like a drum roll: "You must have a team meeting at the halfway point. Halfway through, right at the middle of the contest, you must gather and discuss the strategy for the remaining half of the competition."

Since the start of the contest, exactly H hours, M minutes, and S seconds have elapsed. Your team is extremely nervous. So nervous, in fact, that none of the members can mentally calculate whether the halfway mark has passed or not, so they must program it.

Determine whether more than half of the contest has elapsed, less than half, or if we are exactly at the halfway mark.

Input

A single line containing three integers H, M, and S ($0 \leq H \leq 4$, $0 \leq M \leq 59$, $0 \leq S \leq 59$).

Output

A single line containing a single character: '`+`' to indicate that more than half of the contest has elapsed, '`-`' to indicate that less than half of the contest has elapsed, or '`=`' to indicate that we are exactly at the halfway mark of the contest.

Examples

standard input	standard output
1 2 3	-
3 1 2	+
2 30 0	=

Note

In the first example, one hour, two minutes, and three seconds have elapsed. There is still more than an hour left until the halfway point of the contest.

In the second example, three hours, one minute, and two seconds have elapsed. The halfway point of the contest was more than half an hour ago.

Remember that there are 60 minutes in an hour and 60 seconds in a minute.

Problem B. Breaking Rosarigasino

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Camila is visiting Rosario for the first time. As she walked through its streets, she heard words she didn't understand, so she decided to write them down in her travel log.

Upon arriving at her hotel room, she decided to decipher this mysterious code. After spending hours analyzing the words, she thinks she has figured it out:

People from Rosario take a word, duplicate one of its vowels (`A`, `E`, `I`, `O`, `U`), and then insert the word `GAS` between the two vowels. For example:

$$\texttt{ROSAR}\underline{\texttt{I}}\texttt{NO} \rightarrow \texttt{ROSAR}\underline{\texttt{II}}\texttt{NO} \rightarrow \texttt{ROSAR}\underline{\texttt{IGASI}}\texttt{NO}.$$

Now she wants to take her list and translate it by reversing this transformation. However, she noticed that she had words that did not seem to follow this rule, such as `PRALINE` and `ENTRECOT`. She also found words that could have more than one possible translation, such as `AGASAJOGASO`, which could come from `AGASAJO` or from `AJOGASO`.

Camila wants us to write a program that helps her with this tedious task.

Input

A single line containing a string of N characters ($1 \leq N \leq 100$), consisting of uppercase English letters from `A` to `Z`. It is also guaranteed to contain at least one vowel.

Output

If there is exactly one possible translation, the translated word.

If there is more than one possible translation, the character '`+`'.

If there is no possible translation, the character '`-`'.

Examples

standard input	standard output
ROSARIGASINO	ROSARINO
PRALINE	-
AGASAJOGASO	+
TAGASAP	TAP
ENTREGASEGASEMOS	ENTREGASEMOS

Problem C. Confrontation of Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Two players, Agustín and Brian, play a turn-based game using two arrays of N integers. Agustín has the array $A_1, A_2, \ldots, A_N$, and Brian has the array $B_1, B_2, \ldots, B_N$.

The state of the game is defined by two variables: a current index x and a score s, both initially set to 0. The players take turns alternately, starting with Agustín. On their turn, a player can perform one of the following two actions:

- **Pass:** The player decides not to alter the state of the game.
- **Advance:** The player chooses an index y such that $x < y \leq N$. Upon doing so, the current index is updated to $x = y$, and the score s is completely replaced: it changes to the value A_y if it is Agustín's turn, or to the value B_y if it is Brian's turn.

The game ends immediately when both players decide to pass consecutively. Agustín's goal is to maximize the final value of the score s, while Brian's goal is to minimize it. Your task is to find the final value of s assuming both players play optimally.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$), the length of the arrays.

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($-10^9 \leq A_i \leq 10^9$), Agustín's array.

The third line contains N integers $B_1, B_2, \ldots, B_N$ ($-10^9 \leq B_i \leq 10^9$), Brian's array.

Output

A single integer, the final value of s when both players play optimally.

Examples

standard input	standard output
4 5 -2 -2 3 -1 10 4 8	4
1 -5 12	0

Note

In the first example, Agustín chooses $x = 1$, resulting in $s = A_1 = 5$. After that, Brian chooses $x = 3$ and $s = B_3 = 4$. At that point, Agustín decides to pass, and Brian does as well. These are the optimal moves, and the final score is $s = 4$.

In the second example, it is optimal for both players to pass.

Problem D. Digit One Dilemma

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the middle of the ocean lies a small island called Unoland. The people of this country have a crazy idea in their heads: they believe that the number 1 is the best number in the world.

Naturally, this obsession reached the country's currency as well. One day, the head of the central bank got tired of regular banknotes and issued an order. From that moment on, the bank was only allowed to produce banknotes composed exclusively of the digit 1. As a result, on the island you will only find banknotes of 1 unopeso, 11 unopesos, 111 unopesos, 1111 unopesos, and so on. There is no maximum unopeso banknote; a banknote always exists for any desired number of ones.

The island's population loved this system, as it guarantees that any positive integer can be formed by summing unopeso banknotes, making transactions much easier because you can always pay the exact amount without needing any change. Since carrying a large number of banknotes is annoying, people always seek to minimize the total number of banknotes used when paying. Unoland has hired you to design a program capable of calculating the minimum number of banknotes required to sum up to X unopesos.

Input

A single line containing an integer X ($1 \leq X \leq 10^{100000}$), the total amount of unopesos to be summed.

Output

A single integer, the minimum number of banknotes required to form the value X.

Examples

standard input	standard output
14	4
1997	27
123456789	9
11122233344455566677788899	9

Note

In the first example, we can pay 14 unopesos with 1 banknote of 11 unopesos and 3 banknotes of 1 unopeso.

In the second example, we can pay 1997 unopesos with 1 banknote of 1111 unopesos, 6 banknotes of 111 unopesos, and 20 banknotes of 11 unopesos. This gives a total of $1111 + 6 \times 111 + 20 \times 11 = 1111 + 666 + 220 = 1997$, using $1 + 6 + 20 = 27$ banknotes. It can be shown that we cannot pay this amount using fewer banknotes.

Problem E. Engaging Card Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	1024 megabytes

One day at their grandparents' house, Ana and Beto found a very strange deck of cards. This deck consisted of M cards, each containing an integer, some of which were repeated. Not knowing what the cards were for, they decided to invent a game to pass the time. The rules of the game are as follows:

Ana starts the game by choosing a card from the deck and writing down its number.

Then, they take turns playing alternately, each choosing an unplayed card from the deck such that its sum with the number written down by the other player in the previous turn is a power of two minus one. That is, if the number written down by the other player in the previous turn is x, then the current player must choose a card with a number y such that $x + y = 2^k - 1$ for some integer $k > 0$. Once chosen, they write down the number y.

The game ends when a player cannot choose a card that satisfies the above condition. The player who cannot choose a card on their turn loses the game.

Given the initial deck of cards, determine who will win the game assuming both players play optimally.

Input

The first line contains an integer N ($1 \leq N \leq 10^6$), the number of **distinct** values on the cards in the deck.

The i-th of the following N lines contains two integers A_i and C_i ($1 \leq A_i, C_i \leq 10^9$), indicating that there are C_i cards with the number A_i written on them. The total number of cards in the deck is $M = C_1 + C_2 + \ldots + C_N$.

It is guaranteed that all A_i are pairwise distinct and sorted in strictly increasing order, that is, $A_1 < A_2 < \cdots < A_N$.

Output

A single line containing the name of the winning player if both play optimally: `Ana` if Ana wins, or `Beto` otherwise.

Examples

standard input	standard output
3 2 1 5 2 10 1	Beto
4 1 3 2 1 3 2 4 3	Ana

Note

In the first example, the deck consists of the cards $[2, 5, 5, 10]$. If Ana chooses the card with 10, then Beto chooses one of the cards with 5 (he can do so since $5 + 10 = 15 = 2^4 - 1$). Next, Ana can only choose the card with 2 (since $5 + 2 = 7 = 2^3 - 1$ and $5 + 5 = 10$, which is not a power of two minus one), and Beto chooses the other card with 5. At this point, Ana cannot choose any card because there are no cards left

in the deck, so Beto wins. It can be shown that if Ana chooses a different card on her first turn, Beto also has a winning strategy.

In the second example, the deck consists of the cards $[1, 1, 1, 2, 3, 3, 4, 4, 4]$. Ana can win if she chooses a 1 on her first turn.

Problem F. Flow of Messages

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.75 seconds
Memory limit:	1024 megabytes

In a very close country, Lautaro is in charge of the TUP (Unique Presidential Transmitter), which is responsible for transmitting confidential messages between the government of his country and its neighboring country.

To do this, he has a network of N stations, numbered from 1 to N, where station 1 corresponds to his country's government and station N corresponds to the neighboring country's government. The remaining stations serve the purpose of facilitating message transmission between station 1 and station N.

To enable transmissions, there are M bidirectional links connecting these stations. The i-th of these links allows sending a message between stations A_i and B_i, with a certain delay of w_i seconds that has not yet been determined.

Lautaro is calibrating the delays w_i of this network, which must be integers and sorted in non-decreasing order. That is: $1 \leq w_1 \leq w_2 \leq \ldots \leq w_M$ with $w_i \in \mathbb{Z}$ for $1 \leq i \leq M$.

Lautaro wants to minimize the network's efficiency coefficient, which depends on a parameter k ranging from 1 to M. It is computed using the following formula:

$$\frac{f(w_1, w_2, \ldots, w_M)}{w_1 + w_2 + \ldots + w_k}$$

where $f(w_1, w_2, \ldots, w_M)$ is the minimum time required to send a message from station 1 to station N (possibly using intermediate stations), given that the link delays are $w_1, w_2, \ldots, w_M$.

Since Lautaro does not yet know which value of k will be used to compute this coefficient, he will calculate it for all M integer values of k between 1 and M.

Your task is to help Lautaro and find, for each integer value of k with $1 \leq k \leq M$, the infimum of the network's efficiency coefficient across all possible integer delays satisfying $1 \leq w_1 \leq w_2 \leq \ldots \leq w_M$.

The infimum of a set S is the greatest value c such that $x \geq c$ for all $x \in S$. For example, the infimum of the set $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$ is 0, the infimum of the set $\{x \mid 1 < x < 3\}$ is 1, and the infimum of the set $\{7, 13, 22\}$ is 7. Only in the third set the infimum is also a minimum, since in the other two the infimum does not belong to the set.

Input

The first line contains two integers N and M ($2 \leq N \leq 5000$, $1 \leq M \leq 5000$), the number of stations and the number of links.

The i-th of the following M lines contains two integers A_i and B_i ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$), the stations connected by the i-th link. There may be multiple links connecting the same pair of stations.

It is guaranteed that it is possible to send a message from station 1 to station N, possibly using intermediate stations.

Output

M lines, where the i-th line contains the infimum of the network's efficiency coefficient for $k = i$.

For each value, your output will be accepted if its absolute or relative error is at most 10^{-6}.

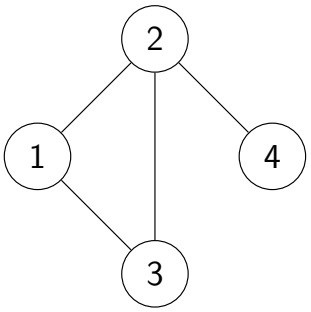
Formally, let your output be a and the jury's answer be b, your answer will be accepted if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Examples

standard input	standard output
4 4 1 2 2 3 3 1 2 4	2 1 0.5 0.3333333
2 1 1 2	1

Note

The following image corresponds to the TUP network for the first example:



For $k = 1$ and $k = 2$, the minimum of the network's efficiency coefficient is achieved with $w_1 = w_2 = w_3 = w_4 = 2$ (which is also the infimum). Here $f(2, 2, 2, 2) = 4$, since it is the minimum time required to send a message from station 1 to station 4 along the path $1 \to 2 \to 4$ with a total delay of $w_1 + w_4 = 2 + 2 = 4$. In the first case, we divide by $w_1 = 2$, and in the second case by $w_1 + w_2 = 2 + 2 = 4$.

For $k = 3$, it can be shown that the infimum is $\frac{1}{2}$, but it is not a minimum because that value cannot be achieved.

Problem G. Galactic Southern Cross

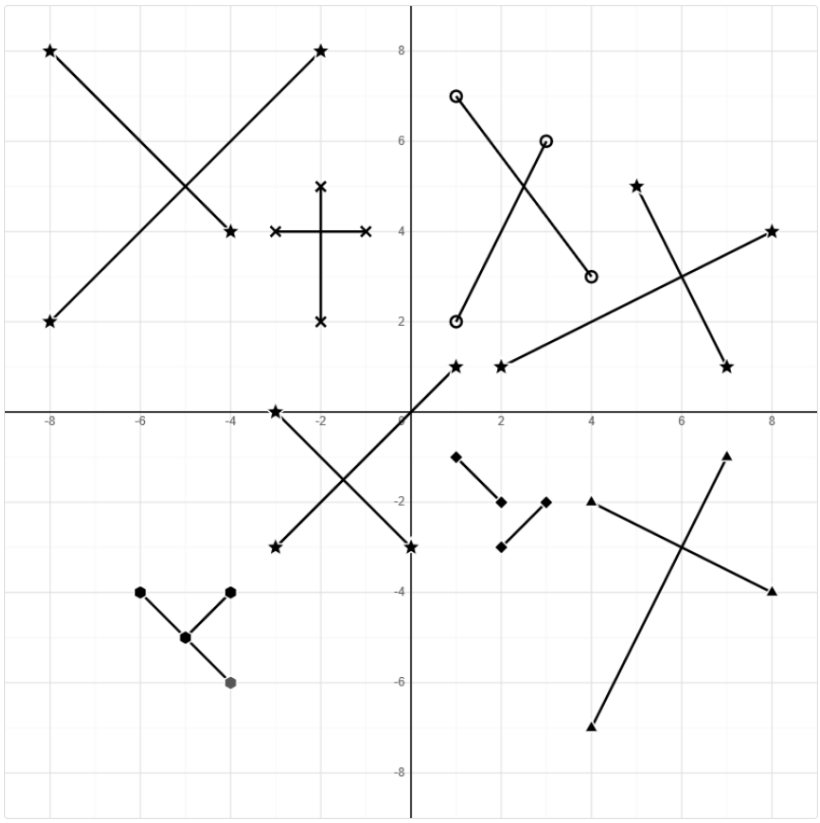
Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

When your ship's Infinite Improbability Drive malfunctions and leaves you stranded on a custom-made planet built by the ancient engineers of Magrathea, it is hard not to panic a bit.

Looking up, you discover an immense artificial night sky filled with stars. You remember that on Earth, people grouped them into constellations, and you immediately identify four stars that remind you of the Southern Cross. Right away, you spot another one, and yet another.

For a set of stars to be considered a "New Southern Cross", it must satisfy all of the following conditions:

- **Four Points**: The constellation consists of exactly four distinct stars.
- **Non-collinearity**: No three stars in this set lie on the same straight line.
- **Perpendicular Intersection**: These four stars can be joined to form two line segments: the shaft and the crossbar, which intersect each other perpendicularly (90°).
- **Bisection**: The intersection point of the segments lies at the midpoint of the crossbar.
- **Alignment**: The stars forming the shaft must be aligned with the South Pole (which is located at $(0, 0)$), or $(0, 0)$ must be a point on that segment.



In the image, we can see examples of valid crosses (which form a New Southern Cross) marked with star vertices (⋆). The others are invalid: crosses (×) and triangles (▲) do not satisfy Alignment, circles (○) and squares (■) do not satisfy Perpendicular Intersection, and filled circles (●) do not satisfy Non-collinearity.

You must count how many sets of stars form a New Southern Cross in the sky you are observing.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$), the number of stars.

The i-th of the following N lines contains two integers X_i and Y_i ($-10^4 \leq X_i, Y_i \leq 10^4$), the coordinates of the i-th star. It is guaranteed that no two stars have the same coordinates.

Output

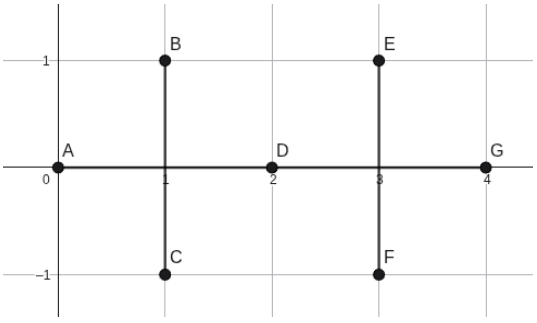
A single integer, the number of sets of stars that form a valid cross.

Examples

standard input	standard output
7 0 0 1 1 1 -1 2 0 3 1 3 -1 4 0	4
6 1 0 3 1 3 -1 4 0 4 2 6 1	1
5 1 0 3 1 3 0 3 -1 4 0	1

Note

In the first example, the stars look like this:



There are four valid crosses, formed by the following sets of stars: $ABCD$, $ABCG$, $AEFG$, and $DEFG$.

Problem H. Harmonic Arrays

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In an array of N positive integers, a coloring is a way of assigning exactly one color to each element of the array.

A coloring will be called prime if there exists a color c such that all numbers painted with color c sum to a prime number.

An array will be called harmonic if all of its colorings are prime, and its elements are between 1 and 10^8.

Given an N, determine whether there exists a harmonic array of size N. If it exists, output such an array.

Input

A line with an integer N ($1 \leq N \leq 13$).

Output

If there does not exist a harmonic array of size N, output a line with the character '`*`'.

Otherwise, output a line with the N elements of the harmonic array $A_1, \ldots, A_N$ ($1 \leq A_i \leq 10^8$).

If there are multiple solutions, any of them will be accepted.

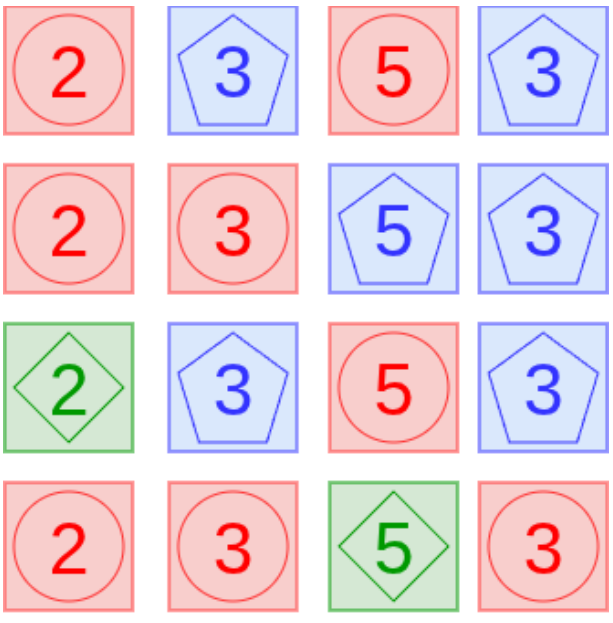
Examples

standard input	standard output
2	29 8
4	2 3 5 3

Note

In the first example, if we paint 29 and 8 with the same color, the only color that exists will have sum $29 + 8 = 37$, which is prime. On the other hand, if we paint 29 and 8 with different colors, the color of 29 will have sum 29, which is prime.

In the second example, some colorings of this array are:



In the first two colorings, the numbers painted red (which have a circle) sum to a prime number.

In the last two colorings, the numbers painted green (which have a diamond) sum to a prime number.

It can be shown that in this array, any other coloring is also prime.

Problem I. Is There a Winner?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A group of N friends has gathered to play a tournament of their favorite video game. The tournament consists of K independent rounds. In each round, exactly one participant wins and receives one point. After all rounds are finished, the player with the highest score is declared champion.

Due to the group's high competitiveness, there is great uncertainty: if, at the end of the rounds, there is no unique winner because of a tie for first place between two or more people, the tournament is considered a failure and ends in sadness.

Given N and K, your task is to determine whether the players can be sure that there will always be a unique champion, regardless of the outcome of each round, or if there is a risk that the tournament will end in a tie.

Input

A line with two integers N and K ($2 \leq N \leq 10^9$, $1 \leq K \leq 10^9$), the number of players and the number of rounds, respectively.

Output

A line with the character '`S`' if the players are guaranteed happiness (that is, it is impossible for there to be a tie for first place), or the character '`N`' if there is a risk of ending in sadness (that is, there exists at least one scenario where a tie for the maximum score occurs).

Examples

standard input	standard output
2 1	S
3 2	N

Note

In the second example, player 1 could win the first round and player 3 the second round. This situation results in a tie for first place and the tournament ends in sadness.

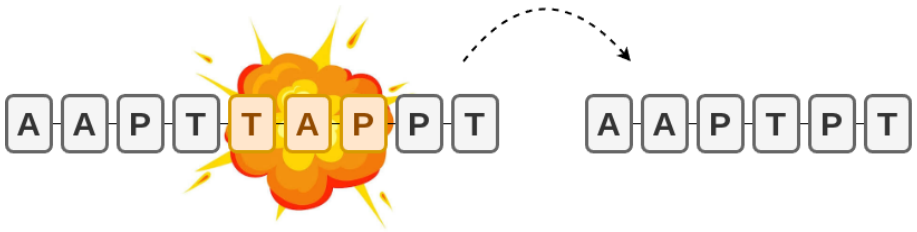
Problem J. Joy of playing TAPum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	1024 megabytes

Juana was given one of the most fun puzzle games ever created: TAPum!

The TAPum game starts with a string of N characters. This string has the same number of '`T`', '`A`', and '`P`' characters, and these are the only characters that appear.

In one move, you may choose three consecutive characters of the string that are pairwise different, and make them... TAPum! Making them TAPum means removing these three consecutive, pairwise different characters from the string. When characters are removed from the string, the remaining ones keep their relative order.



You win the TAPum game if, after zero or more moves, you manage to destroy all N characters of the string, so that none remain.

Determine whether it is possible to win the game or not.

Input

A line with a string of N characters ($3 \leq N \leq 999$).

All characters in the string are '`T`', '`A`', or '`P`', and each of them appears the same number of times.

Output

A line with the character '`S`' to indicate that it is possible to win the game, or '`N`' to indicate that it is not possible.

Examples

standard input	standard output
`AAPTTAPPT`	`S`
`AAAPPPTTT`	`N`

Note

In the first example, the game can be won by making the following moves: `AAPTTAPPT` → `AAPTPT` → `APT` → empty string.

In the second example, no move can be made, so it is impossible to win.

Problem K. Kilograms Exceeded

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Grandfather Laino is getting ready once again to visit another of his dream places, Jujuy.

He already has his suitcase packed and has put on its combination lock, which consists of N dials, each of which contains the digits from 0 to 8 in order.

The lock can only be opened when the correct code is entered, and for that each dial has to show the correct number.

If at some moment a dial shows digit d and we turn it to the right, it changes to digit $d+1$, except when it shows digit 8, in which case it changes to 0. Similarly, if a dial shows digit d and we turn it to the left, it changes to digit $d-1$, except when it shows digit 0, in which case it changes to 8.

Grandfather Laino had set the combination, and to lock the suitcase again he wanted to bring it to the combination with all zeros, so as not to reveal his true code. To achieve this, if a dial showed digit d, he had to turn it d times to the left. Since Grandfather Laino confuses left with right, he turned it d times to the right instead.

When he arrived at the airport, he weighed his suitcase and got a surprise. The suitcase exceeded the maximum allowed weight, so he had to open it to remove some objects. Unfortunately, he forgot the lock combination, but his confusion when turning the dials can help him reconstruct it.

Could you help Grandfather by telling him what his combination was, seeing only how the lock looks after Laino turned the dials?

Input

The first line contains an integer N ($1 \leq N \leq 100$), the number of dials of the lock.

The second line contains N digits $R_1, R_2, \ldots, R_N$ ($0 \leq R_i \leq 8$), where R_i is the digit shown by the i-th dial. It is guaranteed that at least one of the digits is different from 0.

Output

A line with N digits, which form Grandfather Laino's initial combination.

It can be shown that there is a unique valid initial combination.

Examples

standard input	standard output
4 2 0 2 6	1 0 1 3
3 3 8 8	6 4 4

Note

In the first example, Laino's code is `1 0 1 3`. On the first dial, instead of turning it once to the left, he turned it to the right, and that is why it showed a 2. The same happened with the third dial. On the second dial, since it had a zero he should not have turned it and it stayed as it was. The fourth dial had a 3, so he should have turned it 3 times to the left, but he turned it 3 times to the right, showing a 6.

Problem L. Luana's Album

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Luana has a sticker album. The album has M slots for stickers, all in a row. The slots are numbered from 1 to M from left to right. Each sticker also contains a number between 1 and M. There may be multiple stickers with the same number. On the i-th slot, only stickers containing number i can be pasted.

Unfortunately, the album has a very strong restriction for filling it: a sticker can only be pasted if it is to the right of all the stickers already pasted.

Luana buys sticker packs one at a time. Each time she buys a pack, Luana looks at the stickers in the pack in order and, for each one, decides whether to paste it in the album or destroy it forever, always respecting the filling condition. Luana never buys a new pack until she has pasted or destroyed all the stickers from the previous pack.

Unfortunately, all the packs sold by the store are identical: they always contain the same stickers, in the same order. To try to complete the album faster, Luana may turn some packs over before opening them, thus reversing the order in which she looks at the stickers in those packs.

Your task is to tell Luana the maximum number of stickers she can paste in the album and the minimum number of packs she has to buy to paste that many stickers.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 2 \times 10^5$), the number of stickers in each pack and the number of slots in the album.

The second line contains N integers $A_1, \ldots, A_N$ ($1 \leq A_i \leq M$), which indicate the numbers contained in the stickers in the order they appear. The i-th sticker in the pack can only be pasted in the A_i-th slot of the album.

Output

A line with two integers, the first one must be the maximum number of stickers that can be pasted in the album and the second one must be the minimum number of packs Luana has to buy to achieve that.

Examples

standard input	standard output
6 10 2 8 4 7 5 4	5 2
3 3 3 2 1	3 1

Note

In the first example, Luana should buy two packs in order to maximize the number of stickers she pastes in the album:

- First pack: without turning it over, she gets stickers 2, 4 and 5 (discarding stickers 8, 7 and 4, in that order)
- Second pack: she must turn it over, and she gets stickers 7 and 8 (discarding stickers 4, 5, 4 and 2, in that order)

For the second example, Luana buys a single pack and opens it after turning it over.

Problem M. Mendocinos Mismanaging Malbec

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

On a mountain in Mendoza there are N barrels of malbec arranged in a line, numbered from 1 to N from the base of the mountain to the summit. Each barrel has a fixed capacity. At the beginning of a wine season, all barrels are empty.

The season lasts M days. On each day, exactly one of the following operations occurs:

- A winemaker pours V liters of malbec into the B-th barrel. If, after adding the malbec, the amount stored exceeds the barrel's capacity, the excess flows into barrel $B-1$. If that one also exceeds its capacity, the excess continues into barrel $B-2$, and so on. If barrel 1 also overflows, the excess falls to the floor and is lost.
- An oenologist takes all the malbec from all barrels between the L-th barrel and the R-th barrel, inclusive. After this operation, all those barrels become empty.

For each operation in which an oenologist takes malbec, you must say how many liters they obtained.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 2 \times 10^5$), the number of barrels and the number of days in the season, respectively.

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^9$), where A_i is the capacity of the i-th barrel in liters.

Each of the following M lines contains three integers and describes an operation:

- 1 B V ($1 \leq B \leq N$, $1 \leq V \leq 10^9$), indicates that a winemaker pours V liters of malbec into the B-th barrel.
- 2 L R ($1 \leq L \leq R \leq N$), indicates that an oenologist takes all the malbec from the barrels between the L-th and R-th barrel, inclusive.

Output

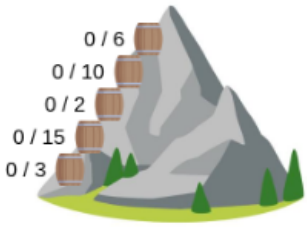
For each operation of the second type, output a line with an integer indicating the number of liters of malbec the oenologist took.

Example

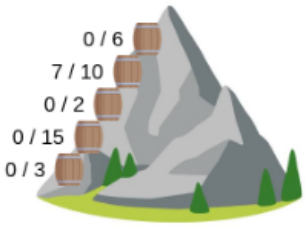
standard input	standard output
5 7 3 15 2 10 6 1 4 7 1 5 10 2 3 4 2 1 5 1 1 5 1 2 12 2 1 2	11 6 15

Note

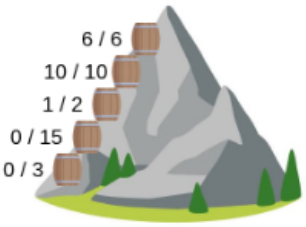
For the example, the following images illustrate how much Malbec there is in each barrel after each operation:



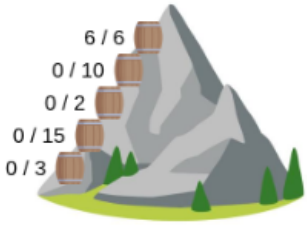
Barrels at the beggining.



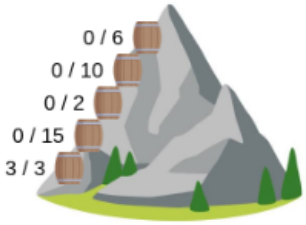
After the first operation.



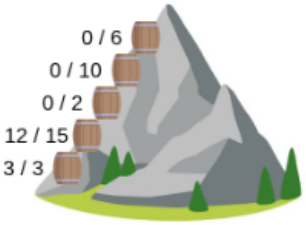
After the second operation.



After the third operation.



After the fifth operation.



After the sixth operation.

The images after the fourth operation and after the last operation were not included since all barrels are empty and look like those in the first image.

Problem N. Number of Strings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given three integers N, K, and A. Your task is to calculate the number of strings of length N that contain exactly K palindromic substrings of length 3, considering that each string is formed using only the first A letters of the alphabet.

A substring is a string that can be obtained from the original string by deleting some characters (possibly zero) from the beginning and some characters (possibly zero) from the end.

Two or more substrings formed by the same letters in the same order, but appearing in different positions of the original string of length N, are **not** counted as the same; instead, each occurrence is counted once. For example, `AAABAAA` has 5 substrings of length 3.

A palindrome is a string that is the same if we read it from left to right or from right to left, such as `SOMOS` or `NEUQUEN`.

Input

A single line with three integers N, K, and A ($1 \leq N \leq 10^6$, $0 \leq K \leq 10^6$, $2 \leq A \leq 26$).

Output

A single integer, the number of strings that satisfy the conditions modulo 998244353.

Examples

standard input	standard output
3 1 2	4
5 2 3	54
2 1 26	0

Note

In the first example, the four strings that satisfy the conditions are `AAA`, `ABA`, `BAB`, and `BBB`. Note that all of them have exactly one palindromic substring of length 3.

In the second example, some strings that satisfy the conditions are `CACAB` or `ABACA`.

In the third example, there are no strings that satisfy all the conditions.